Model selection and clustering in stochastic block models with the exact integrated complete data likelihood

Etienne Côme[a], Pierre Latouche[b]

[a] *Université Paris-Est, IFSTTAR, GRETTIA, F-93166 Noisy-Le-Grand, France*
[b] *Laboratoire SAMM, Université Paris 1 Panthéon-Sorbonne*
90 rue de Tolbiac, F-75634 Paris Cedex 13, France

Abstract

The stochastic block model (SBM) is a mixture model used for the clustering of nodes in networks. It has now been employed for more than a decade to analyze very different types of networks in many scientific fields such as Biology and social sciences. Because of conditional dependency, there is no analytical expression for the posterior distribution over the latent variables, given the data and model parameters. Therefore, approximation strategies, based on variational techniques or sampling, have been proposed for clustering. Moreover, two SBM model selection criteria exist for the estimation of the number K of clusters in networks but, again, both of them rely on some approximations. In this paper, we show how an analytical expression can be derived for the integrated complete data log likelihood. We then propose an inference algorithm to maximize this exact quantity. This strategy enables the clustering of nodes as well as the estimation of the number clusters to be performed at the same time and no model selection criterion has to be computed for various values of K. The algorithm we propose has a better computational cost than existing inference techniques for SBM and can be employed to analyze large networks with ten thousand nodes. Using toy and true data sets, we compare our work with other approaches.

Keywords: Random graphs, stochastic block models, integrated classification likelihood.

1. Introduction

There is a long history of research on networks which goes back to the earlier work of Moreno [1]. Because they are simple data structures yet capable of representing

complex systems, they are used in many scientific fields [2, 3]. Originally considered in social sciences [4] to characterize relationships between actors [5, 6], networks are also used to describe neural networks [7], powergrids [8], and the Internet [9, 10]. Other examples of real networks can be found in Biology with the use of regulatory networks to describe the regulation of genes by transcriptional factors [11] or metabolic networks to represent pathways of biochemical reactions [12]. As the number of networks used in practice has been increasing, a lot of attention has been paid on developing graph clustering algorithms to extract knowledge from their topology. Existing methods usually aim at uncovering very specific patterns in the data, namely communities or disassortative mixing. For an exhaustive review, we refer to [13].

Most graph clustering algorithms look for communities, where two nodes of the same community are more likely to be connected than nodes of different communities. These techniques [14, 15] often maximize the modularity score proposed by Girvan and Newman [16] for clustering. However, recent work of Bickel and Chen [17] showed that they were asymptotically biased and tended to lead to the discovery of an incorrect community structure, even for large graphs. Alternative strategies, see for instance [18], are generally related to the probabilistic model of Handcock, Raftery and Tantrum [19] which generalizes the work of Hoff, Raftery and Handcock [20]. Nodes are first mapped into a a latent space and then clustered depending on their latent positions. Community structure algorithms are commonly used for affiliation network analysis. As mentioned in [21], other graph clustering algorithms aim at uncovering dissasortative mixing in networks where, contrary to community structure, nodes mostly connect to nodes of different clusters. They are particularly suitable for the analysis of bipartite or quasi bipartite networks [22].

In this paper, we consider the stochastic block model (SBM) proposed by Nowicki and Snijders [23] which is a probabilistic generalization [4, 5] of the work of White, Boorman and Breiger [24]. As pointed out by Daudin, Picard and Robin [25], SBM can be seen as a mixture model for graphs. It assumes that nodes are spread into K clusters and uses a $K \times K$ matrix $\mathbf{\Pi}$ to describe the connection probabilities between pairs of nodes. No assumption is made on $\mathbf{\Pi}$ such that very different

structures can be taken into account. In particular, as shown in [26], contrary to the methods mentioned previously, SBM can be used to retrieve both communities and disassortative mixing in networks.

Many extensions have been developed to overcome some limits of the standard SBM. For example, Mariadassou, Robin and Vacher [27] introduced recently a probabilistic framework to deal with valued edges, allowing covariates to be taken into account. While the first model they proposed explains the value of an edge, between a pair of nodes, through their clusters only, the second and third approaches do account for covariates through Poisson regression models. This framework is relevant in practice because extra information on the edges is sometimes available, such as phylogenetic distances in host-parasite networks or amounts of energy transported between nodes in powergrids.

Another drawback of SBM is that it assumes that each node belongs to a single cluster while many objects in real world applications belong to several groups or communities [28]. To tackle this issue Airoldi *et al.* [29] proposed the mixed membership stochastic block model (MMSBM) [30, 31]. A latent variable $\boldsymbol{\pi}_i$, drawn from a Dirichlet distribution, is associated to each node i of a network. Given a pair (i, j) of nodes, two binary latent vectors $\mathbf{Z}_{i\to j}$ and $\mathbf{Z}_{i\leftarrow j}$ are then considered. The vector $\mathbf{Z}_{i\to j}$ is assumed to be sampled from a multinomial distribution with parameters $(1, \boldsymbol{\pi}_\mathbf{i})$ and describes the cluster membership of i in its relation towards j. By symmetry, $\mathbf{Z}_{i\leftarrow j}$ is drawn from multinomial distribution with parameters $(1, \boldsymbol{\pi}_\mathbf{j})$ and characterizes the cluster membership of j in its relation towards i. Thus, in MMSBM, since each node can have different latent vectors through its relations towards other nodes, it can belong to several clusters. The connection probability between i and j is finally given by $p_{ij} = \mathbf{Z}_{i\to j}^\intercal \mathbf{B} \mathbf{Z}_{i\leftarrow j}$. The overlapping stochastic block model (OSBM) was proposed by Latouche, Birmelé and Ambroise [28] as an alternative probabilistic model for networks allowing overlapping clusters. Contrary to MMSBM, edges are influenced by the fact that some nodes belong to multiple clusters. Thus, each node i is characterized by a binary latent vector $\mathbf{Z}_i$ sampled from a product of Bernoulli distributions. An edge between nodes i and j is then drawn from a Bernoulli distribution with parameter $p_{ij=g(a_{\mathbf{Z}_i,\mathbf{Z}_j})}$. The function $g(\cdot)$

is the logistic function while $a_{\mathbf{Z}_i,\mathbf{Z}_j}$ is a real variable describing the interactions between the nodes, depending on the different clusters they are associated with. It is given by $a_{\mathbf{Z}_i,\mathbf{Z}_j} = \mathbf{Z}_i^\intercal \mathbf{W} \mathbf{Z}_j + \mathbf{Z}_i^\intercal \mathbf{U} + \mathbf{V}^\intercal \mathbf{Z}_j + W^*$. Finally, we mention the work of Karrer and Newman [32] who proposed an interesting extension of SBM to deal with node degree heterogeneity inside clusters. The model deals with valued edges and includes another set of parameters describing vertices attractiveness. Using the right constraints the model is identifiable (up to permutations of clusters) and the atractivity parameters can be directly related to vertices degree. This work was extended to oriented networks in [33] and finally tools for model selection between different models are derived in [34].

In this paper, we wont consider the extensions for SBM we mentionned, we will rather focus on the standard SBM. Our goal here is not to propose new extensions, allowing a SBM like model to be applicable on specific types of networks, or to introduce new latent structures. Conversely, considering the standard SBM, which has been widely used in practice for network analysis, for more than a decade, we aim at developping a new optimization procedure, improving over existing inference strategies. In SBM, the posterior distribution over the latent variables, given the parameters and the observed data, cannot be factorized due to conditional dependency. Therefore, optimization techniques such as the expectation maximization (EM) algorithm cannot be used directly for clustering. To tackle this issue, Daudin, Picard and Robin [25] proposed an approximation method based on a variational EM algorithm. Note that an online version of this algorithm exists [35]. A Bayesian framework was also considered by Nowicki and Snijders [23] where conjugate priors for the model parameters were introduced. Again, because the posterior distribution over the model parameters, given the data, is not tractable, approximation techniques were employed for inference. Thus, Nowicki and Snijders [23] used a Gibbs sampling procedure while Latouche, Birmelé and Ambroise [36] relied on a variational Bayes EM algorithm. To our knowledge, only two model selection criteria, the integrated classification likelihood (ICL) and the integrated likelihood variational Bayes (ILvb) have been developed for SBM in order to estimate the number K of clusters in networks. Standard criteria such as the Akaike information

criterion (AIC) or bayesian information criterion (BIC) cannot be used because they rely on the SBM observed data log likelihood which is not tractable in practice (see for instance [26]). ICL was originally developed by Biernacki, Celeux and Govaert [37] for Gaussian mixture models and then adapted by Daudin, Picard and Robin [25] to SBM. It is based on Laplace and Stirling approximations of the integrated complete data log likelihood. As shown in [38], it tends to miss some important structures present in the data for small data sample, because of the asymptotic approximations. To tackle this drawback, Latouche, Birmelé and Ambroise proposed in [36] the ILvb criterion which relies on a variational Bayes approximation of the integrated observed data log likelihood.

In this paper, we show how an analytical expression of the integrated complete data log likelihood can be obtained in a Bayesian framework and that no asymptotic approximation is required. We call the corresponding criterion ICL_{ex} where ex stands for exact. We then propose a greedy inference algorithm which maximizes this exact quantity. The strategy has three advantages compared to existing approaches. First, it maximizes an analytical criterion directly derived from SBM, while variational techniques for instance rely on lower bounds for approximation. Thus, the lower bound of the variational EM algorithm proposed by Daudin, Picard and Robin [25] approximates the observed data log likelihood, while Latouche, Birmelé and Ambroise [36] introduced a lower bound to estimate the integrated observed data log likelihood. Second, ICL_{ex} only depends on all the latent binary vectors $\mathbf{Z}_i$, stored in the matrix $\mathbf{Z}$, and the number K of clusters, not on the model parameters which are marginalized out. Therefore, the optimization task focus on $(K, \mathbf{Z})$ and is purely combinatorial. When using the Gibbs algorithm [23], the successive samples for $\mathbf{Z}$ and model parameters are highly correlated. As a consequence, nodes tend to be stuck in clusters, after a few iterations. Similar remarks could be made for the variational EM and variational Bayes EM algorithms. In our case, because the parameters are marginalized out (collapsed) the method is expected to explore more easily the latent space of $\mathbf{Z}$. This property is at the core of collapsing methods (for more details, we refer to [39]). Finally, our strategy enables the clustering of nodes as well as the estimation of the number of clusters to be performed

at the same time and no model selection criterion has to be computed for various values of K. Starting from a complex model with $K = K_{up}$ clusters, (K_{up} being an upper bound for K), the proposed algorithm swaps labels until ICL_{ex} reaches a local maximum. During the process, clusters may disappear, *i.e.* their cardinality reaches zero. Such an approach leads to a simple and time attractive algorithm with complexity of $\mathcal{O}(L + NK_{up}^2)$, with L the total number of edges in the network and N the number of vertices.

As we shall see through a series of experiments, the greedy algorithm takes benefit of computing the exact ICL and improves over existing methods, both in terms of clustering and model selection. It can also deal with large networks with tens of thousands of vertices.

2. The stochastic block model

We consider a binary network with N nodes represented by its adjacency matrix $\mathbf{X}$ such that $X_{ij} = 1$ if there is an edge from node i to node j, 0 otherwise. In this paper, we focus on directed networks, *i.e.* relations are oriented. Therefore $\mathbf{X}$ is not symmetric. Moreover, we do not consider any self loop, that is an edge from a node to itself. We emphasize that all the optimization equations derived in this work can easily be adapted to deal with undirected networks or to take into account self loops.

2.1. Model and notations

The stochastic block model (SBM) introduced by Nowicki and Snijders [23] assumes that the nodes are spread into K clusters with connectivity patterns described by a $K \times K$ matrix $\mathbf{\Pi}$. The cluster of each node is given by its binary membership vector $\mathbf{Z}_i$ sampled from a multinomial distribution :

$$\mathbf{Z}_i \sim \mathcal{M}(1, \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)), \sum_{k=1}^{K} \alpha_k = 1,$$

such that $Z_{ik} = 1$ if i belongs to cluster k and zero otherwise. Contrary to the work of Latouche, Birmelé and Ambroise [28], each node belongs to a single cluster, that

is $\sum_{k=1}^{K} Z_{ik} = 1, \forall i$. Given the vectors $\mathbf{Z}_i$ and $\mathbf{Z}_j$, an edge between node i and j is then drawn from a Bernoulli distribution with probability Π_{kl} :

$$X_{ij}|Z_{ik}Z_{jl} = 1 \sim \mathcal{B}(\Pi_{kl}).$$

This leads to a simple yet flexible generative model for networks. First, all the vectors $\mathbf{Z}_i$ are sampled independently. We denote $\mathbf{Z}$ the binary $N \times K$ matrix storring the $\mathbf{Z}_i$s as raw vectors :

$$p(\mathbf{Z}|\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\alpha}) = \prod_{i=1}^{N} \prod_{k=1}^{K} \alpha_k^{Z_{ik}}. \tag{1}$$

Then, given the latent structure $\mathbf{Z}$, all the edges in $\mathbf{X}$ are drawn independently :

$$\begin{aligned} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\Pi}) &= \prod_{i \neq j}^{N} p(X_{ij}|\mathbf{Z}_i, \mathbf{Z}_j, \boldsymbol{\Pi}) \\ &= \prod_{i \neq j}^{N} \prod_{k,l}^{K} \mathcal{B}(X_{ij}; \Pi_{kl})^{Z_{ik}Z_{jl}} \\ &= \prod_{i \neq j}^{N} \prod_{k,l}^{K} \left(\Pi_{kl}^{X_{ij}} (1 - \Pi_{kl})^{1 - X_{ij}} \right)^{Z_{ik}Z_{jl}}. \end{aligned} \tag{2}$$

2.2. Integrated classification likelihood criteria

In this paper, we consider the integrated complete data log likelihood $\log p(\mathbf{X}, \mathbf{Z}|K)$ in order to focus on the inference of $\mathbf{Z}$ and K from the observed data $\mathbf{X}$, all the SBM parameters $(\boldsymbol{\alpha}, \boldsymbol{\Pi})$ being integrated out. We first recall existing approximations and then show in Section 2.2.2 how an analytical expression of this quantity can be derived.

2.2.1. Asymptotic ICL criterion

When considering a factorized prior distribution $p(\boldsymbol{\alpha}, \boldsymbol{\Pi}|K) = p(\boldsymbol{\alpha}|K)p(\boldsymbol{\Pi}|K)$ over the model parameters, as in [37], the integrated complete data log likelihood easily decomposes into two terms:

$$\begin{aligned} \log p(\mathbf{X}, \mathbf{Z}|K) &= \log \left(\int_{\boldsymbol{\alpha}, \boldsymbol{\Pi}} p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\Pi}, \boldsymbol{\alpha}|K) d\boldsymbol{\alpha} d\boldsymbol{\Pi} \right) \\ &= \log \left(\int_{\boldsymbol{\Pi}} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\Pi}, K) p(\boldsymbol{\Pi}|K) d\boldsymbol{\Pi} \int_{\boldsymbol{\alpha}} p(\mathbf{Z}|\boldsymbol{\alpha}, K) p(\boldsymbol{\alpha}|K) d\boldsymbol{\alpha} \right) \\ &= \log p(\mathbf{X}|\mathbf{Z}, K) + \log p(\mathbf{Z}|K). \end{aligned} \tag{3}$$

However, for an arbitrary choice of the priors $p(\boldsymbol{\alpha}|K)$ and $p(\boldsymbol{\Pi}|K)$, the marginal distributions $p(\mathbf{X}|\mathbf{Z}, K)$ as well as $p(\mathbf{Z}|K)$ are usually not tractable and (3) does not have any analytical form. To tackle this issue, Daudin, Picard and Robin [25] proposed an asymptotic approximation of $\log p(\mathbf{X}, \mathbf{Z}|K)$, so called integrated classification likelihood criterion (ICL). Note that ICL was originally proposed by Biernacki, Celeux and Govaert [37] for Gaussian mixture models. It was then adapted by Biernacki, Celeux and Govaert [38] to mixtures of multivariate multinomial distributions and to the SBM model by Daudin, Picard and Robin [25]. In the case we consider of a directed graph without self-loop, ICL is given by:

$$\begin{aligned} ICL(\mathbf{Z}, K) &\approx \log p(\mathbf{X}, \mathbf{Z}|K) \\ &= \max_{\boldsymbol{\alpha},\boldsymbol{\Pi}} \log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\alpha}, \boldsymbol{\Pi}, K) - \frac{1}{2}K^2 \log\left(N(N-1)\right) - \frac{K-1}{2}\log(N). \end{aligned}$$

For an extensive description of the use of Laplace and Stirling approximations to derive the ICL criterion, we refer to [37]. Since it approximates the integrated complete data log likelihood, ICL is known to be particularly suitable when the focus is on the clustering task and not on the estimation of the data density. However, as shown in [38, 27], it tends to miss some important structures present in the data because of the (asymptotic) approximations.

We emphasize that ICL is only used in the literature as a model selection criterion. In practice, a clustering method such as an EM like algorithm for instance is employed to obtained some estimates $\tilde{\mathbf{Z}}$ of $\mathbf{Z}$, for various values of the number K of classes. ICL is then computed for every pair $(\tilde{\mathbf{Z}}, K)$ and the pair $(\tilde{\mathbf{Z}}^*, K^*)$ is chosen such that the criterion is maximized. Thus, ICL is optimized only through the results $(\tilde{\mathbf{Z}}, K)$ produced by the clustering algorithm. Conversely, after having given an analytical expression ICL_{ex} of the integrated complete data log likelihood in the next section, we will show in Section 3 how to optimize directly ICL_{ex} with respect to $\mathbf{Z}$ and K.

2.2.2. Exact ICL criterion

We rely on the same Bayesian framework as in [23] and [26]. Thus, we consider non informative conjugate priors for the model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Pi}$. Since $\boldsymbol{\alpha}$, describing the cluster proportions, parametrizes a multinomial distribution (1), we

rely on a Dirichlet prior distribution:

$$p(\boldsymbol{\alpha}) = \mathrm{Dir}\left(\boldsymbol{\alpha};\ \mathbf{n}^0 = (n_1^0, \ldots, n_K^0)\right).$$

A common choice consists in fixing the hyperparameters to 1/2, *i.e.* $n_k^0 = 1/2, \forall k$. Such a distribution corresponds to a non informative Jeffreys prior which is known to be proper [40]. A uniform distribution can also be obtained by setting the hyperparameters to 1.

Moreover, since the presence or absence of an edge between nodes is sampled from a Bernoulli distribution, we consider independent Beta prior distributions to model the connectivity matrix $\mathbf{\Pi}$:

$$p(\mathbf{\Pi}) = \prod_{k,l}^{K} \mathrm{Beta}(\Pi_{kl};\ \eta_{kl}^0, \zeta_{kl}^0).$$

Again, if no prior information is available, all hyperparameters η_{kl}^0 and ζ_{kl}^0 can be set to 1/2 or 1 to obtain a Jeffreys or uniform distribution.

With these choices of conjugate prior distributions over the model parameters, the marginal distributions $p(\mathbf{X}|\mathbf{Z}, K)$ as well as $p(\mathbf{Z}|K)$ in (3) have analytical forms, and so has the integrated complete data log likelihood, as proved in AppendixA. We call ICL_{ex} the corresponding criterion, where ex stands for exact. It is given by:

$$\begin{aligned} ICL_{ex}(\mathbf{Z}, K) &= \log p(\mathbf{X}, \mathbf{Z}|K) \\ &= \sum_{k,l}^{K} \log\left(\frac{\Gamma(\eta_{kl}^0 + \zeta_{kl}^0)\Gamma(\eta_{kl})\Gamma(\zeta_{kl})}{\Gamma(\eta_{kl} + \zeta_{kl})\Gamma(\eta_{kl}^0)\Gamma(\zeta_{kl}^0)}\right) + \log\left(\frac{\Gamma(\sum_{k=1}^{K} n_k^0)\prod_{k=1}^{K}\Gamma(n_k)}{\Gamma(\sum_{k=1}^{K} n_k)\prod_{k=1}^{K}\Gamma(n_k^0)}\right), \end{aligned}$$

where the components n_k are:

$$n_k = n_k^0 + \sum_{i=1}^{N} Z_{ik}, \forall k \in \{1, \ldots, K\},$$

and can be seen as pseudo counters of the number of nodes in each class. Moreover, the parameters (η_{kl}, ζ_{kl}) are given by:

$$\eta_{kl} = \eta_{kl}^0 + \sum_{i \neq j}^{N} Z_{ik} Z_{jl} X_{ij}, \forall (k, l) \in \{1, \ldots, K\}^2,$$

and

$$\zeta_{kl} = \zeta_{kl}^0 + \sum_{i \neq j}^{N} Z_{ik} Z_{jl} (1 - X_{ij}), \forall (k, l) \in \{1, \ldots, K\}^2.$$

They represent pseudo counters of the number of edges and non-edges connecting nodes of class k to nodes of class l, respectively.

Note that the ICL_{ex} criterion is related to the variational Bayes approximation of the integrated observed data log likelihood $\log p(\mathbf{X}|K)$ proposed by Latouche, Birmelé and Ambroise [36]. The key difference is that the parameters $(n_k, \eta_{kl}, \zeta_{kl})$ in ICL_{ex} depend on the hard assignment $\mathbf{Z}$ of nodes to classes and not on approximated posterior probabilities $\boldsymbol{\tau}$. Moreover, ICL_{ex} does not involve any entropy term as in [36].

3. Greedy optimization

Since the model parameters have been marginalized out, the ICL_{ex} criterion only involves the cluster indicator matrix $\mathbf{Z}$ whose dimensionality depends on the number K of clusters. Thus, this integrated likelihood is only a function of a partition $\mathcal{P}$, *i.e.* an assignment of the vertices to clusters. Looking directly for a global maximum of ICL_{ex} is not feasible because it involves testing every possible partition of the vertices with various values of K. However, this is a combinatorial problem for which heuristics exist to obtain local maxima. In this paper, we rely on greedy heuristics which have been shown to scale well with sample sizes [14]. These approaches have already been used for graph clustering using *ad-hoc* criteria such as modularity [14, 41] and are reminiscent of the well known iterated conditional modes algorithm of Besag [42] used for maximum *a posteriori* estimation in Markov random fields.

The algorithm (see Algorithm 1) starts with a SBM model with $K = K_{up}$ clusters, K_{up} being an upper bound for the number of clusters. K_{up} is assumed to be given as an input along with a $N \times K_{up}$ matrix $\mathbf{Z}$. In practice, K_{up} is set to a large value using user knowledge on the problem at hand, while $\mathbf{Z}$ can be initialized with the methods described in the next section. The algorithm then cycles randomly through all the vertices of the network. At each step, a single node i is considered while all the membership vectors $\mathbf{Z}_j$ for $j \neq i$ are hold fixed. If i is currently in cluster g, the method looks for every possible label swapping, *i.e.* removing i from cluster g and assigning it to a cluster $h \neq g$, and computes the corresponding change $\Delta_{g \to h}$ in the ICL_{ex} criterion. Note that $\Delta_{g \to h}$ takes two forms (see AppendixB) whether

cluster g is empty after removing i or not. If no label swapping induces an increase of the criterion, the vector $\mathbf{Z}_i$ is not modified. Otherwise, the label swapping with the maximal increase is applied and $\mathbf{Z}_i$ is changed accordingly. During the process, clusters may disappear, *i.e.* their cardinality reaches zero. Each time one of these moves is accepted, the model is updated and the corresponding column is removed from the cluster indicator matrix $\mathbf{Z}$. Finally, the algorithm stops if a complete pass over the vertices did not lead to any increase of the ICL_{ex} criterion. Thus, the algorithm, automatically infers the number of clusters while clustering the vertices of the network. Starting with an over-segmented initial solution our approach simplifies the model until a local maximum is reached.

3.1. Complexity

In order to set up such an algorithm, it is sufficient to know how to compute the changes in the ICL_{ex} criterion induced by the possible swap movements (from cluster g to cluster h) for a given node i, the others being kept fixed. Such changes can be computed efficiently (see AppendixB for details) and the complexity of finding the best swap movement for a node is in average $\mathcal{O}(l+K^2)$, where l is the average number of edges per node. Such complexity can be achieved, since good approximations of the logarithm of the gamma function are available with constant running time. The greedy algorithm has therefore a total complexity of $\mathcal{O}(N(l + K_{up}^2) + L)$, since a swap movement cost is $\mathcal{O}(l + K^2)$; the initialization of the edges counters (η_{kl}, ζ_{kl}) cost is L (the total number of edges in the graph) and several complete passes over the set of nodes will be performed (typically less than 10). Eventually, this can be simplified in $\mathcal{O}(NK_{up}^2 + L)$ and compared to the complexity of $\mathcal{O}(LK_{up}^3)$ achieved using a variational algorithm and a model selection criterion as in [25, 36]. Indeed, contrary to our approach which estimates the number of clusters in a single run, while clustering the nodes, these approaches are run multiple times for various values of K and K^* is chosen such that the corresponding model selection criterion is maximized. Since each run costs $\mathcal{O}(LK^2)$, the overall complexity is $\mathcal{O}(LK_{up}^3)$.

Algorithm 1: Greedy ICL

Set $K = K_{up}$; stop $= 0$;

Initialize the $N \times K_{up}$ matrix $\mathbf{Z}$; Compute $\boldsymbol{\eta}$,$\boldsymbol{\zeta}$,$\mathbf{n}$;

while stop $\neq 1$ **do**

  $V = \{1, \ldots, N\}$; stop $= 1$;

  while V *not empty* **do**

    Select a node i randomly in V ; Remove i from V ;

    If i is in cluster g, compute all terms $\Delta_{g \to h}, \forall h \neq g$;

    if *at least one* $\Delta_{g \to h}$ *is positive* **then**

      stop $= 0$;

      Find h such that $\Delta_{g \to h}$ is maximum ;

      Swap labels of i : $Z_{ig} = 0$ and $Z_{ih} = 1$;

      if g *is empty* **then**

        Remove column g in $\mathbf{Z}$; Set $K = K - 1$;

      end

      Update rows and columns (g, h) of the matrices $\boldsymbol{\eta}$ and $\boldsymbol{\zeta}$;

      Update the components g and h of vector $\mathbf{n}$;

    end

  end

end

Result: $(\mathbf{Z}, K)$

3.2. Initialization and restarts

Several solutions are possible for initializing the algorithm, a simple choice consisting in sampling random partitions while a more relevant though expensive starting point can be obtained with the k-means algorithm. One possible trade-off in terms of computational burden is to use only few iterations of k-means. We used the latter method in all the experiments that we carried out. Moreover, since our method is only guaranteed to reach a local optima, a common strategy is to run the optimization algorithm with multiple initializations and to keep the best one according to the ICL_{ex} criterion.

3.3. Hierarchical clustering

Eventually, in a final step, it is possible to check that merge movements between clusters do not induce any increase of the objective function. This can be done with a greedy hierarchical algorithm which costs $\mathcal{O}(K^3)$ (see details in AppendixC). Since the labels swap algorithm usually greatly reduces the number of clusters ($K << K_{up}$), the computational cost of this last step is low.

Such a scheme leads to a fast algorithm: sparse networks take about 15 seconds for $N = 500$ nodes, and about five minutes for $N = 5000$ with a naive Matlab implementation.

4. Experiments on synthetic data

To assess the greedy optimization method, a simulation study was performed and the proposed solution was compared with available implementations of algorithms for SBM inference:

- **vbmod**, [43], a variational-based approach dedicated to the search of community structures, implemented in Matlab and C. The random graph model they considered can be seen as a constrained SBM where all terms on the diagonal of the connectivity matrix $\mathbf{\Pi}$ are set to a unique parameter λ and off-diagonal terms to another parameter ϵ,

- **mixer**, [25], another variational approach but one which can deal with all types of SBM models (not only communities structures) implemented in R and C,

- **colsbm**, [44], a collapsed Gibbs sampler for SBM in C. The last version of the code is used in this experimental section. It involves an additional move type compared to the algorithm described in the associated publication. This move was found to greatly enhance the results.

Our goal here is to evaluate the ability of the different solutions to recover a simulated clustering *without* knowing the number of clusters. Only a reasonable upper bound K_{up} on K will be provided to the algorithms when needed. We recall

that the variational methods optimize a lower bound for various values of K and select K^* such that a model selection criterion is maximized: ICL for **mixer** and ILvB for **vbmod**. Conversely, the collapsed Gibbs sampler automatically provides an estimate of K, since the posterior of K is made available.

The performances are assessed in terms of normalized mutual information (see for instance [45]) between the estimated cluster membership matrix $\mathbf{Z}^e$ and the simulated one $\mathbf{Z}^s$. The mutual information $I(\mathbf{Z}^e, \mathbf{Z}^s)$ between two partitions is to this end defined by:

$$I(\mathbf{Z}^e, \mathbf{Z}^s) = \sum_{k,l}^{K} p_{kl} \log\left(\frac{p_{kl}}{p_k^e p_l^s}\right), \tag{4}$$

with

$$p_{kl} = \frac{1}{N}\sum_{i,j}^{N} Z_{ik}^e Z_{jl}^s,\ p_k^e = \frac{1}{N}\sum_{i=1}^{N} Z_{ik}^e,\ p_l^s = \frac{1}{N}\sum_{i=1}^{N} Z_{il}^s.$$

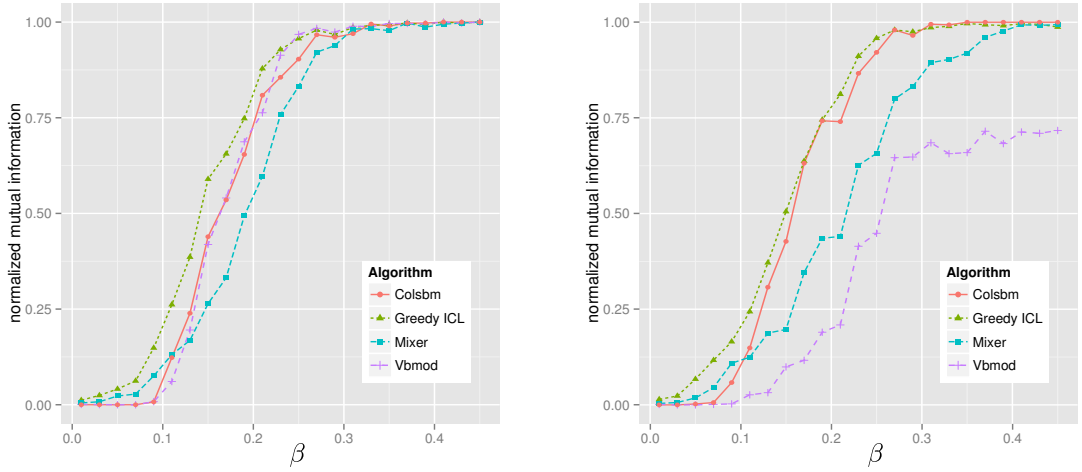
The measure $I(\mathbf{Z}^e, \mathbf{Z}^s)$ describes how much is learnt about the true partition if the estimated one is known, and *vice versa*. The mutual information is not an ideal similarity measure when the two partitions have a different number of clusters and it is therefore preferable to use a normalized version of the mutual information such as:

$$NI(\mathbf{Z}^e, \mathbf{Z}^s) = \frac{I(\mathbf{Z}^e, \mathbf{Z}^s)}{\max\left(H(\mathbf{Z}^e), H(\mathbf{Z}^s)\right)}, \tag{5}$$

with $H(\mathbf{Z}) = -\sum_{k=1}^{K} p_k \log(p_k)$ and $p_k = \frac{1}{N}\sum_i^N Z_{ik}$. The performances are evaluated on simulated clustering problems of varying complexity and with different settings, in order to give insights about the influence of the number K of clusters, of the number of vertices N and of the type of connectivity matrix $\mathbf{\Pi}$.

4.1. Setting 1: small scale community structures

The first setting is a classical community simulation with $N = 100$ vertices and $K = 5$ clusters. The cluster proportions are set to $\boldsymbol{\alpha} = (1/5, 1/5, 1/5, 1/5, 1/5)$ and the connectivity matrix takes a diagonal form with off-diagonal elements equal to 0.01: $\mathbf{\Pi}_{kl} = 0.01, \forall k \neq l$ and diagonal elements given by $\mathbf{\Pi}_{kk} = \beta, \forall k$. β is a complexity tuning parameter which ranges from 0.45 to 0.01. When β reaches 0.01, the model is not identifiable (the connectivity matrix is constant) and the true cluster memberships cannot be recovered. The set of simulated problems is therefore of



Setting 1 Setting 2

Figure 1: Mean of mutual information between estimated and true cluster membership matrices using 20 simulated graphs for each value of β in $\{0.45, 0.43, \ldots, 0.03, 0.01\}$, and with $N = 100, K = 5, \epsilon = 0.01$ for the different algorithms **greedy icl**, **vbmod**, **colsbm** and **mixer**.

varying complexity: from problems with a clear structure when $\beta = 0.45$ to problems without any structure when $\beta = 0.01$. The experiments are performed twenty times for each value of β and the average of the normalized mutual information over these twenty simulated graphs is depicted in Figure 1 (left) for all the algorithms. In order to produce results as comparable as possible, the parameters of the different algorithms were set as follows: **vbmod, mixer** and **greedy icl** were all started ten times and for each method the best run was selected according to the corresponding model selection criterion. The variational methods were run with K between 2 and 20 and the best clustering kept as a final result. For **greedy icl**, the parameters of the prior η^0, ζ^0 and n_k^0 were set to 1 and K_{up} fixed to twenty. Finally the collapsed Gibbs sampler was run for 250 000 iterations (more than twice the default value).

The results illustrated in Figure 1 show that **greedy icl** outperforms the other methods for complex problems, *i.e.* low values of β. The simulated clustering is recovered until β reaches 0.25. Above this value the different algorithms perform identically, but beyond this limit the results of **greedy icl** are a little bit better. During the transition **greedy icl** gets slightly better results than the other algorithms, it is followed by **colsbm** and **vbmod** which give close results and **mixer** that deviates earlier from the planted clustering.

4.2. Setting 2: small scale community structures with a hub cluster

The second setting aims at exploring the performances of the methods when the latent structure does not correspond only to communities. To this end, graphs were generated using the stochastic block model with affiliation probability matrix $\mathbf{\Pi}$ of the form as in [36]:

$$\mathbf{\Pi} = \begin{pmatrix} \beta & \beta & \dots & \dots & \beta \\ \beta & \beta & \epsilon & \dots & \epsilon \\ \beta & \epsilon & \beta & \dots & \epsilon \\ \beta & \epsilon & \dots & \beta & \epsilon \\ \beta & \epsilon & \dots & \dots & \beta \end{pmatrix}.$$

The clusters correspond therefore to communities, except one cluster of hubs which connects with probability β to all other clusters. Graphs with $N = 100$ vertices, $K = 5$ clusters and $\boldsymbol{\alpha} = (1/5, 1/5, 1/5, 1/5, 1/5)$ were generated using this connection pattern. The parameter ϵ was set to 0.01 and β ranged as previously from 0.45 to 0.01. Eventually, the other simulation parameters did not change. The results are shown in Figure 1 (right).

As expected, the **vbmod** algorithm, which looks only for communities, is strongly affected by this change of setting and systematically misses the hub cluster. For the remaining methods, the best results are achieved by **greedy icl** which still uncovers the planted clustering when $\beta > 0.25$, whereas **mixer** starts to drop at β equals 0.4. The collapsed Gibbs sampler achieves also good results in this setting, very close to those of **greedy icl** and out-performs **mixer**.

4.3. Setting 3: medium scale community structures

The third setting is a classical community simulation but with more nodes and clusters, in order to study the effect of these two parameters. Thus, the number of vertices was set to $N = 500$ and the number of clusters to $K = 10$. The cluster proportions were defined as $\boldsymbol{\alpha} = (1/10, \dots, 1/10)$ and all the other parameters kept the same value as previously. For this third experiment, the results presented in Figure 2 are identical for **greedy icl**, **colsbm** and **mixer**. The **vbmod** algorithm seems to be more affected by the dimensionality of the problem, and did not recover

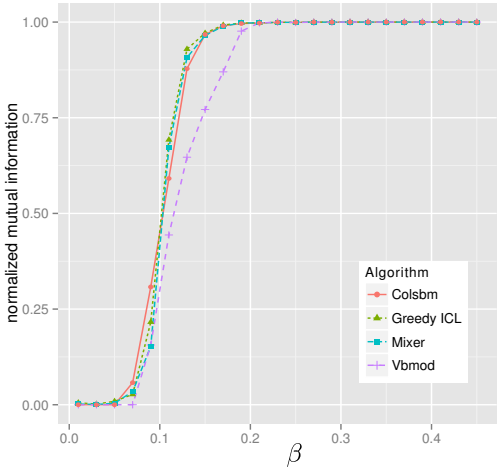


Figure 2: Mean of mutual information between estimated and true cluster membership matrices using 20 simulated graphs for each value of β in $\{0.45, 0.43, \ldots, 0.03, 0.01\}$, and with $N = 500, K = 10, \epsilon = 0.01$ for the different algorithms **greedy icl**, **vbmod**, **colsbm** and **mixer**.

exactly the true clusters when β is under 0.2. The results obtained by the different algorithms in this setting are better than those obtained previously. This can easily be explained by the increase in the number of nodes per cluster. The transitions between high and low values of the normalized mutual information were also sharper than in the previous experiments, for the same reasons.

4.4. Setting 4: large scale problem with complex structure

The final setting involves larger graphs with $N = 10\,000$ vertices. The planted structure is also not a purely community pattern. Some interactions between clusters are activated randomly using a Bernoulli distribution as described by the following generative model:

$$\Pi_{kl} = \begin{cases} ZU + (1 - Z)\epsilon, \text{ if } k \neq l \\ U, \text{ if } k = l \end{cases} \tag{6}$$

with $Z \sim \mathcal{B}(0.1)$, $U \sim \mathcal{U}(0.45)$ and $\epsilon = 0.01$. The size of the problem and the complex nature of the underlying structure, let only two algorithms able to deal with these graphs namely **greedy icl** and **colsbm**, since **mixer** cannot handle such large graphs and **vbmod** deals only with community structure. Both were used to cluster 20 simulated graphs generated using this scheme. The greedy algorithm was started

using $K_{up} = 100$ and the same setting as previously for the prior distributions. The results presented as boxplots in Figure 3 give a clear advantage to **greedy icl** over the collapsed sampler. Thus, **greedy icl** achieves an average normalized mutual information of 0.88 whereas **colsbm** reaches only 0.67. In fact, the greedy solution ended with around 80 clusters for all the simulations whereas the Gibbs sampler gives more than 240 clusters in average and therefore produces highly over segmented partitions of the graphs.

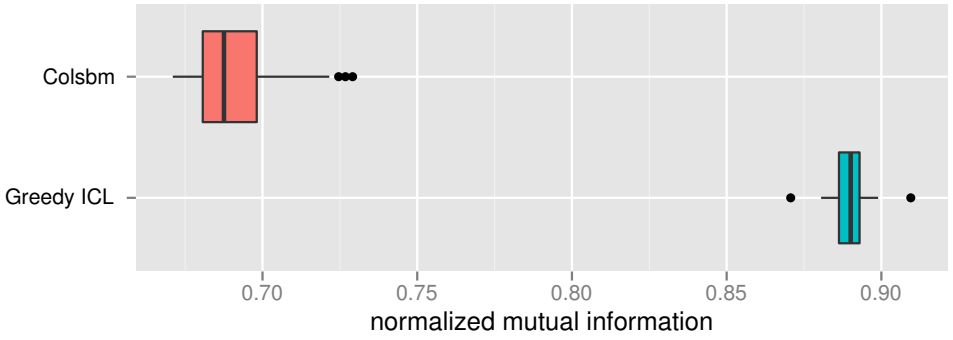


Figure 3: Mean of the mutual information between estimated and true cluster membership matrices using 20 simulated graphs with $N = 10000$ and $K = 50$.

To summarize the results we obtained in all the experiments we carried out, it appears that **greedy icl** compares favourably with the other existing solutions for SBM, in all the settings. The results obtained in complex setting, *i.e.* large graphs and a complex underlying structure (Setting 4) are particularly encouraging since **greedy icl** clearly outperforms the collapsed Gibbs sampler.

5. Real dataset: communities of blogs

The proposed algorithm was finally tested on a real network where vertices correspond to blogs and edges to known hyperlinks between the blogs. All the blogs considered are related to a common topic, *i.e.* illustrations and comics.

The network was built using a community extraction procedure [46] which starts from known seeds and expands them to find a dense core of nodes surrounding them. It is made of 1360 blogs linked by 33 805 edges. The data set is expected to present specific patterns, namely communities, where two blogs of the same community

are more likely to be connected that nodes of different communities. To test this hypothesis we used the greedy ICL algorithm and did a qualitative comparison of the results with those obtained with the community discovery method of Blondel *et al.* [41].

Starting with $K_{up} = 100$ clusters, greedy ICL found $K = 37$ clusters. The corresponding clusters are illustrated in Figure 4 which is an image of the adjacency matrix with rows/columns sorted by cluster number. Thus, it appears that the clusters found correspond in their vast majority to small sub-communities. These sub-communities all correspond to known groups. For instance a group of blogs of illustrators for Disney was found. Other examples include clusters of blogs of students who went to the same illustration school such as the ECMA school of Angouleme or the "Gobelins École de l'image". However, some clusters have more complex connectivity structures and are made of hubs which highly connect to blogs of different clusters. They correspond to blogs of famous writers such as Boulet.

To give a qualitative idea of the interest of the found clustering, we also give the results obtained by the community discovery algorithm of Blondel *et al.* [41] in Figure 5. With this approach only 8 clusters are found, corresponding all to sub-communities. Clusters of hubs could not be recovered. The major difference between the number of clusters estimated by the two methods may be explained by two facts. Firstly, modularity is known to be prone to a resolution limit problem [47] which prevents such a solution to extract small scale structures. This explains why the small sub-community extracted by **greedy icl** are not recovered using the modularity. For the time being, the behaviour of the ICL_{ex} criterion with respect to the resolution limit problem is not clear and will deserve further investigations. However, we notice that on this dataset finer structures than those obtained using modularity are recovered. Secondly, the difference in the way the two criteria use degree correction or not [32] can also explain the disparity in the number of clusters. While modularity is a degree-corrected criterion which downscales the weights of the edges between highly connected vertices, the ICL_{ex} criterion for the basic stochastic block model used here is not. Using a degree correction or not is a modelling choice which deserves to be validated and investigated; however, it seems that even without

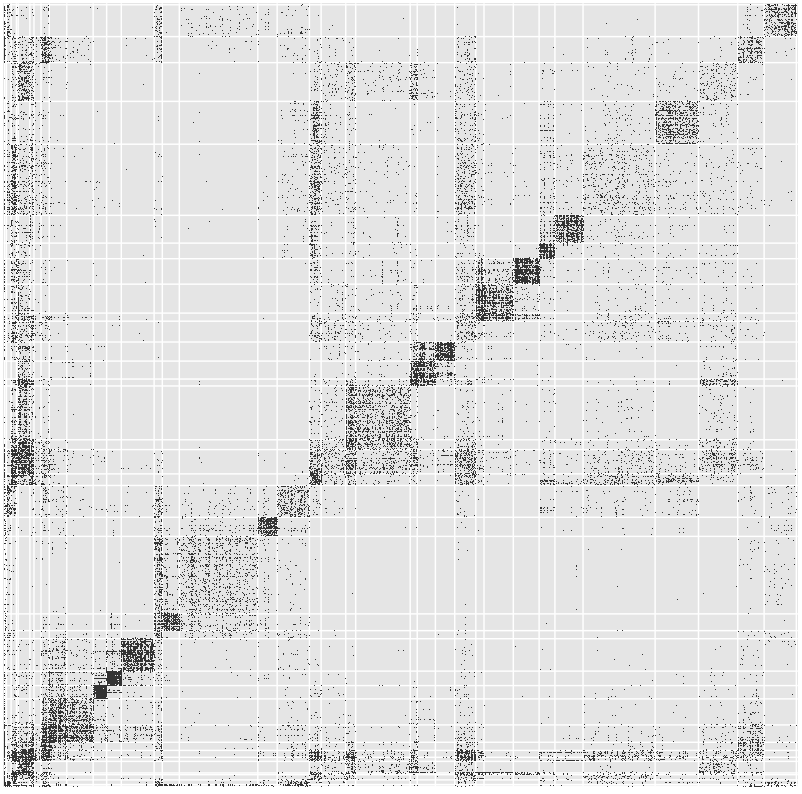


Figure 4: Adjacency matrix of the network of blogs, the rows/columns are sorted by cluster number with clusters found by the greedy ICL algorithm. The cluster boundaries are depicted with white lines.

degree correction the results obtained by **greedy icl** are meaningful, the hub clusters being interesting *per se*.

6. Conclusion

In this paper, we showed how an analytical expression of the integrated complete data log likelihood could be derived using conjugate priors for the model parameters, and that no asymptotic approximations were required. We then proposed a greedy optimization algorithm to maximize this exact quantity. Starting from an over segmented partition, the approach simplifies the model, while clustering the vertices, until a local maximum is reached. This greedy algorithm has a competitive complexity and may handle networks with tens of thousands of vertices. We illustrated on simulated data that the method improves over existing graph clustering algorithms, both in terms of model selection and clustering of the vertices. A quali-

Figure 5: Adjacency matrix of the network of blogs, the rows/columns are sorted by cluster number with clusters found by modularity optimization. The clusters boundaries are depicted with white lines.

tative comparison between methods was also carried out on an original network we built from blogs related to illustration, comics, and animations.

We emphasize that the methodology we considered can be adapted to other mixture models. In particular, we will investigate the case of the degree corrected stochastic block model which have been shown to give promising results on real data.

APPENDIX

AppendixA. Marginal distributions

Proposition AppendixA.1. *The marginal distribution $p(\mathbf{Z}|K)$ is given by:*

$$p(\mathbf{Z}|K) = \frac{C(\mathbf{n})}{C(\mathbf{n}^0)},$$

where the components of the vector $\mathbf{n}$ are $n_k = n_k^0 + \sum_{i=1}^N Z_{ik}$, for all k in $\{1, \dots, K\}$ and the function $C(\cdot)$ is such that $C(\mathbf{x}) = \frac{\prod_{k=1}^K \Gamma(x_k)}{\Gamma(\sum_{k=1}^K x_k)}$ for all $\mathbf{x}$ in $\mathbb{R}^K$.

Proof.

$$\begin{aligned} p(\mathbf{Z}|\boldsymbol{\alpha}, K)p(\boldsymbol{\alpha}|K) &= \left(\prod_{i=1}^N \prod_{k=1}^K \alpha_k^{Z_{ik}}\right) \mathrm{Dir}(\boldsymbol{\alpha}; \mathbf{n}^0) \\ &= \left(\prod_{k=1}^K \alpha_k^{\sum_{i=1}^N Z_{ik}}\right) \frac{1}{C(\mathbf{n}^0)} \prod_{k=1}^K \alpha_k^{n_k^0 - 1} \\ &= \frac{1}{C(\mathbf{n}^0)} \prod_{k=1}^K \alpha_k^{n_k^0 - 1 + \sum_{i=1}^N Z_{ik}} \\ &= \frac{1}{C(\mathbf{n}^0)} \prod_{k=1}^K \alpha_k^{n_k - 1}, \end{aligned} \tag{A.1}$$

and we denote $\mathbf{n}$ the vector with components $n_k = n_k^0 + \sum_{i=1}^N Z_{ik}$ for all k in $\{1, \dots, K\}$. Thus

$$\begin{aligned} p(\mathbf{Z}|\boldsymbol{\alpha}, K)p(\boldsymbol{\alpha}|K) &= \frac{C(\mathbf{n})}{C(\mathbf{n}^0)} \frac{1}{C(\mathbf{n})} \prod_{k=1}^K \alpha_k^{n_k - 1} \\ &= \frac{C(\mathbf{n})}{C(\mathbf{n}^0)} \mathrm{Dir}(\boldsymbol{\alpha}; \mathbf{n}). \end{aligned}$$

Therefore

$$\begin{aligned} p(\mathbf{Z}|K) &= \int_{\boldsymbol{\alpha}} p(\mathbf{Z}|\boldsymbol{\alpha}, K)p(\boldsymbol{\alpha}|K)d\boldsymbol{\alpha} \\ &= \frac{C(\mathbf{n})}{C(\mathbf{n}^0)} \int_{\boldsymbol{\alpha}} \mathrm{Dir}(\boldsymbol{\alpha}; \mathbf{n})d\boldsymbol{\alpha} \\ &= \frac{C(\mathbf{n})}{C(\mathbf{n}^0)}. \end{aligned}$$

□

Proposition AppendixA.2. *The marginal distribution $p(\mathbf{X}|\mathbf{Z}, K)$ is given by:*

$$p(\mathbf{X}|\mathbf{Z}, K) = \prod_{k,l}^K \frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)},$$

where $\eta_{kl} = \eta_{kl}^0 + \sum_{i \neq j}^N Z_{ik} Z_{jl} X_{ij}$ *and* $\zeta_{kl} = \zeta_{kl}^0 + \sum_{i \neq j}^N Z_{ik} Z_{jl}(1 - X_{ij})$ *for all* (k, l) *in* $\{1, \ldots, K\}^2$. *The function* $B(a, b)$ *is such that* $B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ *for all* (a, b) *in* $\mathbb{R}^2$.

Proof.

$$\begin{aligned}
p(\mathbf{X}|\mathbf{Z}, \mathbf{\Pi}, K)p(\mathbf{\Pi}|\mathbf{K}) &= \left(\prod_{i \neq j}^{N} \prod_{k,l}^{K} \left(\Pi_{kl}^{X_{ij}} (1 - \Pi_{kl})^{1 - X_{ij}} \right)^{Z_{ik} Z_{jl}} \right) \prod_{k,l}^{K} \text{Beta}(\Pi_{kl}; \eta_{kl}^0, \zeta_{kl}^0) \\
&= \left(\prod_{k,l}^{K} \Pi_{kl}^{\sum_{i \neq j}^N Z_{ik} Z_{jl} X_{ij}} (1 - \Pi_{kl})^{\sum_{i \neq j}^N Z_{ik} Z_{jl}(1 - X_{ij})} \right) \\
&\times \prod_{k,l}^{K} \frac{1}{B(\eta_{kl}^0, \zeta_{kl}^0)} \Pi_{kl}^{\eta_{kl}^0 - 1} (1 - \Pi_{kl})^{\zeta_{kl}^0 - 1} \\
&= \prod_{k,l}^{K} \frac{1}{B(\eta_{kl}^0, \zeta_{kl}^0)} \Pi_{kl}^{\eta_{kl}^0 - 1 + \sum_{i \neq j}^N Z_{ik} Z_{jl} X_{ij}} (1 - \Pi_{kl})^{\zeta_{kl}^0 - 1 + \sum_{i \neq j}^N Z_{ik} Z_{jl}(1 - X_{ij})} \\
&= \prod_{k,l}^{K} \frac{1}{B(\eta_{kl}^0, \zeta_{kl}^0)} \Pi_{kl}^{\eta_{kl} - 1} (1 - \Pi_{kl})^{\zeta_{kl} - 1},
\end{aligned}$$

where $\eta_{kl} = \eta_{kl}^0 + \sum_{i \neq j}^N Z_{ik} Z_{jl} X_{ij}$ and $\zeta_{kl} = \zeta_{kl}^0 + \sum_{i \neq j}^N Z_{ik} Z_{jl}(1 - X_{ij})$ for all (k, l) in $\{1, \ldots, K\}^2$. Thus

$$\begin{aligned}
p(\mathbf{X}|\mathbf{Z}, \mathbf{\Pi}, K)p(\mathbf{\Pi}|K) &= \prod_{k,l}^{K} \frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)} \frac{1}{B(\eta_{kl}, \zeta_{kl})} \Pi_{kl}^{\eta_{kl} - 1} (1 - \Pi_{kl})^{\zeta_{kl} - 1} \\
&= \prod_{k,l}^{K} \frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)} \text{Beta}(\Pi_{kl}; \eta_{kl}, \zeta_{kl}).
\end{aligned}$$

Therefore

$$\begin{aligned}
p(\mathbf{X}|\mathbf{Z}, K) &= \int_{\mathbf{\Pi}} p(\mathbf{X}|\mathbf{Z}, \mathbf{\Pi}, K)p(\mathbf{\Pi}|K) d\mathbf{\Pi} \\
&= \int_{\mathbf{\Pi}} \left(\prod_{k,l}^{K} \frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)} \text{Beta}(\Pi_{kl}; \eta_{kl}, \zeta_{kl}) \right) d\mathbf{\Pi} \\
&= \prod_{k,l}^{K} \left(\frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)} \int_{\Pi_{kl}} \text{Beta}(\Pi_{kl}; \eta_{kl}, \zeta_{kl}) d\Pi_{kl} \right) \\
&= \prod_{k,l}^{K} \frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)}.
\end{aligned}$$

□

Proposition AppendixA.3. *Using factorized and conjugate prior distributions over the model parameters, the integrated complete data log likelihood is given by:*

$$\log p(\mathbf{X}, \mathbf{Z}|K) = \sum_{k,l}^{K} \log\left(\frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)}\right) + \log\left(\frac{C(\mathbf{n})}{C(\mathbf{n}^0)}\right),$$

where

- $\eta_{kl} = \eta_{kl}^0 + \sum_{i\neq j}^{N} Z_{ik}Z_{jl}X_{ij}$ *for all* (k,l) *in* $\{1,\ldots,K\}^2$
- $\zeta_{kl} = \zeta_{kl}^0 + \sum_{i\neq j}^{N} Z_{ik}Z_{jl}(1-X_{ij})$ *for all* (k,l) *in* $\{1,\ldots,K\}^2$
- *the components of the vector* $\mathbf{n}$ *are* $n_k = n_k^0 + \sum_{i=1}^{N} Z_{ik}$*, for all* k *in* $\{1,\ldots,K\}$
- *the function* $B(a,b)$ *is such that* $B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ *for all* (a,b) *in* $\mathbb{R}^2$
- *the function* $C(\cdot)$ *is such that* $C(\mathbf{x}) = \frac{\prod_{k=1}^{K}\Gamma(x_k)}{\Gamma(\sum_{k=1}^{K} x_k)}$ *for all* $\mathbf{x}$ *in* $\mathbb{R}^K$

Proof. Considering factorized prior distributions, the integrated complete data log likelihood decomposes into two terms:

$$\begin{aligned}
\log p(\mathbf{X}, \mathbf{Z}|K) &= \log\left(\int_{\boldsymbol{\alpha},\boldsymbol{\Pi}} p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\Pi}, \boldsymbol{\alpha}|K) d\boldsymbol{\alpha} d\boldsymbol{\Pi}\right) \\
&= \log\left(\int_{\boldsymbol{\Pi}} p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\Pi}, K) p(\boldsymbol{\Pi}|K) d\boldsymbol{\Pi} \int_{\boldsymbol{\alpha}} p(\mathbf{Z}|\boldsymbol{\alpha}, K) p(\boldsymbol{\alpha}|K) d\boldsymbol{\alpha}\right) \quad \text{(A.2)} \\
&= \log p(\mathbf{X}|\mathbf{Z}, K) + \log p(\mathbf{Z}|K).
\end{aligned}$$

Using Propositions AppendixA.1 and AppendixA.2 in (A.2) gives:

$$\log p(\mathbf{X}, \mathbf{Z}|K) = \sum_{k,l}^{K} \log\left(\frac{B(\eta_{kl}, \zeta_{kl})}{B(\eta_{kl}^0, \zeta_{kl}^0)}\right) + \log\left(\frac{C(\mathbf{n})}{C(\mathbf{n}^0)}\right).$$

□

AppendixB. Change in ICL induced by a swap movement $i : g \to h$

At each step of the greedy ICL algorithm, a single node i is considered. If i is currently in cluster g, the method tests every possible label swapping $g \to h$, that is removing i from cluster g and assigning it to a cluster $h \neq g$. The corresponding changes in the ICL_{ex} criterion are denoted $\Delta_{g\to h}$. In order to derive the calculation of each term $\Delta_{g\to h}$, for all $h \neq g$, we consider two cluster indicator matrices $\mathbf{Z}$ as

well as $\mathbf{Z}^{test}$. $\mathbf{Z}$ describes the current partition of the vertices in the network, while $\mathbf{Z}^{test}$ represents the partition after applying the swap $g \to h$:

$$\begin{cases} \mathbf{Z}_j^{test} = \mathbf{Z}_j, \forall j \neq i \\ \\ Z_{ik}^{test} = Z_{ik} = 0, \forall k \neq g, h \end{cases}$$

while

$$\begin{cases} Z_{ig}^{test} = 0, Z_{ig} = 1 \\ \\ Z_{ih}^{test} = 1, Z_{ih} = 0 \end{cases}$$

Thus

$$\Delta_{g \to h} = ICL_{ex}(\mathbf{Z}^{test}, K^{test}) - ICL_{ex}(\mathbf{Z}, K).$$

Note that $\Delta_{g \to h}$ takes two forms whether cluster g is empty after removing i or not. In the later scenario, the model dimensionality changes ($K^{test} = K - 1$) and this must be taken into account to evaluate the possible increase induced by the swap movement.

AppendixB.1. Case 1 : $\sum_i Z_{ig}^{test} > 0$. Cluster g not empty after removing i

$$\begin{aligned}
\Delta_{g \to h} &= \log\left(\frac{C(\mathbf{n}^{test})}{C(\mathbf{n})}\right) + \sum_{k,l}^{K} \log\left(\frac{B(\eta_{kl}^{test}, \zeta_{kl}^{test})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&= \log\left(\frac{\Gamma(n_g^{test})\Gamma(n_h^{test})}{\Gamma(n_g)\Gamma(n_h)}\right) + \sum_{l=1}^{K} \sum_{k \in \{g,h\}} \log\left(\frac{B(\eta_{kl}^{test}, \zeta_{kl}^{test})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&+ \sum_{k \notin \{g,h\}} \sum_{l \in \{g,h\}} \log\left(\frac{B(\eta_{kl}^{test}, \zeta_{kl}^{test})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&= \log\left(\frac{\Gamma(n_g - 1)\Gamma(n_h + 1)}{\Gamma(n_g)\Gamma(n_h)}\right) + \sum_{l=1}^{K} \sum_{k \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&+ \sum_{k \notin \{g,h\}} \sum_{l \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&= \log\left(\frac{n_h}{n_g - 1}\right) + \sum_{l=1}^{K} \sum_{k \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right) \\
&+ \sum_{k \notin \{g,h\}} \sum_{l \in \{g,h\}} \log\left(\frac{B(\eta_{kl} + \delta_{kl}^{(i)}, \zeta_{kl} + \rho_{kl}^{(i)})}{B(\eta_{kl}, \zeta_{kl})}\right),
\end{aligned}$$

with $\delta_{kl}^{(i)}$ the changes in edges counter η_{kl} induced by the label swap:

$$\begin{aligned}\delta_{kl}^{(i)} &= \mathbb{1}_{\{k=h\}} \sum_{j\neq i}^{N} Z_{jl} X_{ij} + \mathbb{1}_{\{l=h\}} \sum_{j\neq i}^{N} Z_{jk} X_{ji} - \mathbb{1}_{\{k=g\}} \sum_{j\neq i}^{N} Z_{jl} X_{ij} \\ &- \mathbb{1}_{\{l=g\}} \sum_{j\neq i}^{N} Z_{jk} X_{ji}.\end{aligned}$$

Moreover, $\rho_{kl}^{(i)}$ is defined in the following:

$$\rho_{kl}^{(i)} = \left(\mathbb{1}_{\{k=h\}} - \mathbb{1}_{\{k=g\}}\right)(n_l - n_l^0 - Z_{il}) + \left(\mathbb{1}_{\{l=h\}} - \mathbb{1}_{\{l=g\}}\right)(n_k - n_k^0 - Z_{ik}) - \delta_{kl}^{(i)}.$$

These update quantities can be computed in $O(l_i)$ with l_i the degree of i (total number of edges from and to i). Therefore the complexity of finding the best swap movement for a node is $\mathcal{O}(l + K^2)$, l for computing the $\delta_{kl}^{(i)}$ and K^2 to compute the Δ_{swap} with all the possible h labels and keep the best one.

AppendixB.2. Case 2 : $\sum_i Z_{ig}^{test} = 0$, cluster g disappear

In this case the dimensionality of $\mathbf{n}^0$ changes and we will denote by $\mathbf{n}^{0*} = (n^0, \ldots, n^0)$ the corresponding vector of size $K-1$.

$$\begin{aligned}\Delta_{g\to h} &= \log\left(\frac{C(\mathbf{n}^0)}{C(\mathbf{n})}\frac{C(\mathbf{n}^{test})}{C(\mathbf{n}^{0*})}\right) \\ &+ \sum_{\substack{(k,l)\neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl}+\delta_{kl}^{(i)}, \zeta_{kl}+\rho_{kl}^{(i)})}{B(\eta_{kl},\zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0,\zeta_{kl}^0)}{B(\eta_{kl},\zeta_{kl})}\right) \\ &= \log\left(\frac{n_h}{n^0}\frac{\Gamma\left((K-1)n^0\right)\Gamma(Kn^0+N)}{\Gamma(K\,n^0)\Gamma((K-1)n^0+N)}\right) \\ &+ \sum_{\substack{(k,l)\neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl}+\delta_{kl}^{(i)}, \zeta_{kl}+\rho_{kl}^{(i)})}{B(\eta_{kl},\zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0,\zeta_{kl}^0)}{B(\eta_{kl},\zeta_{kl})}\right)\end{aligned}$$

the complexity in this case is the same as previously *i.e.* $\mathcal{O}(l + K^2)$.

AppendixC. Change in ICL induced by a merge movement

$$
\begin{aligned}
\Delta_{g\cup h} &= \log\left(\frac{C(\mathbf{n}^0)}{C(\mathbf{n})}\frac{C(\mathbf{n}^{test})}{C(\mathbf{n}^{0*})}\right) \\
&+ \sum_{\substack{(k,l)\neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl}+\delta_{kl}^{(i)},\zeta_{kl}+\rho_{kl}^{(i)})}{B(\eta_{kl},\zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0,\zeta_{kl}^0)}{B(\eta_{kl},\zeta_{kl})}\right) \\
&= \log\left(\Gamma(n^0)\frac{\Gamma\left((K-1)n^0\right)\Gamma(Kn^0+N)}{\Gamma(K\,n^0)\Gamma((K-1)n^0+N)}\frac{\Gamma(n_h+n_g-n^0)}{\Gamma(n_g)\Gamma(n_h)}\right) \\
&+ \sum_{\substack{(k,l)\neq g \\ k=h \text{ or } l=h}} \log\left(\frac{B(\eta_{kl}+\delta_{kl}^{(i)},\zeta_{kl}+\rho_{kl}^{(i)})}{B(\eta_{kl},\zeta_{kl})}\right) + \sum_{k=g \text{ or } l=g} \log\left(\frac{B(\eta_{kl}^0,\zeta_{kl}^0)}{B(\eta_{kl},\zeta_{kl})}\right)
\end{aligned}
$$

with $\delta_{kl}^{(i)}$ the changes in edges counter η_{kl} induced by the merge:

$$
\delta_{kl}^{(i)} = \mathbb{1}_{\{k=h\}}(\eta_{gl}-\eta_{gl}^0) + \mathbb{1}_{\{l=h\}}(\eta_{kg}-\eta_{kg}^0) + \mathbb{1}_{\{k=h \text{ and } l=h\}}(\eta_{gg}-\eta_{gg}^0). \quad \text{(C.1)}
$$

Moreover, $\rho_{kl}^{(i)}$ is defined in the following:

$$
\rho_{kl}^{(i)} = \mathbb{1}_{\{k=h\}}(\zeta_{gl}-\zeta_{gl}^0) + \mathbb{1}_{\{l=h\}}(\zeta_{kg}-\zeta_{kg}^0) + \mathbb{1}_{\{k=h \text{ and } l=h\}}(\zeta_{gg}-\zeta_{gg}^0). \quad \text{(C.2)}
$$

[1] J. L. Moreno, Who shall survive?: a new approach to the problem of Human interrelations, Nervous and Mental Disease Publishing, Washington DC, 1934.

[2] A. L. Barabási, Z. N. Oltvai, Network biology: understanding the cell's functional organization, Nature Rev. Genet 5 (2004) 101–113.

[3] G. Palla, A. Barabási, T. Vicsek, Quantifying social group evolution, Nature 446 (2007) 664–667.

[4] S. Fienberg, S. Wasserman, Categorical data analysis of single sociometric relations, Sociological Methodology 12 (1981) 156–192.

[5] J. Holland, K. Laskey, S. Leinhard, Stochastic block models : First steps, Social Networks 5 (1993) 109–137.

[6] R. Boulet, B. Jouve, F. Rossi, N. Villa, Batch kernel som and related laplacian methods for social network analysis, Neurocomputing 71 (7–9) (2008) 1257–1273.

[7] J. G. White, E. Southgate, J. N. Thompson, S. Benner, The structure of the nervous system of the nematode caenorhabditis elegans, Philosophical Transactions. Royal Society London B 314 (1986) 1–340.

[8] D. J. Watts, S. H. Strogatz, Collective dynamics of small-world networks, Nature 393 (1998) 440–442.

[9] L. Adamic, N. Glance, The political blogosphere and the 2004 us election, in: WWW Workshop on the Weblogging Ecosystem, 2005.

[10] H. Zanghi, C. Ambroise, V. Miele, Fast online graph clustering via erdos-renyi mixture, Pattern Recognition 41 (12) (2008) 3592–3599.

[11] R. Milo, S. Shen-Orr, S. Itzkovitz, D. Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, Science 298 (2002) 824–827.

[12] V. Lacroix, C. Fernandes, M.-F. Sagot, Motif search in graphs: pplication to metabolic networks, Transactions in Computational Biology and Bioinformatics 3 (2006) 360–368.

[13] A. Goldenberg, A. Zheng, S. Fienberg, E. Airoldi, A survey of statistical network models, Foundations and Trends in Machine Learning 2 (2) (2010) 129–233.

[14] M. E. J. Newman, Fast algorithm for detecting community structure in networks, Physical Review Letter E 69 (2004) 0066133.

[15] M. E. J. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences 103 (2006) 8577–8582.

[16] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, in: Proceedings of the National Academy of Sciences, Vol. 99, 2002, pp. 7821–7826.

[17] P. J. Bickel, A. Chen, A non parametric view of network models and newman-girvan and other modularities, Proceedings of the National Academy of Sciences 106 (2009) 21068–21073.

[18] P. N. Krivitsky, M. S. Handcock, A. E. Raftery, P. D. Hoff, Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models, Social Networks 31 (2009) 204–213.

[19] M. S. Handcock, A. E. Raftery, J. M. Tantrum, Model-based clustering for social networks, Journal of the Royal Statistical Society 170 (2007) 1–22.

[20] P. Hoff, A. Raftery, M. Handcock, Latent space approaches to social network analysis, Journal of the Royal Statistical Society 97 (2002) 1090–1098.

[21] M. Newman, E. Leicht, Mixture models and exploratory analysis in networks, Proceedings of the National Academy of Sciences 104 (2007) 9564–9569.

[22] E. Estrada, J. A. Rodriguez-Velazquez, Spectral measures of bipartivity in complex networks, Physical Review E 72 (2005) 046105.

[23] K. Nowicki, T. A. B. Snijders, Estimation and prediction for stochastic blockstructures, Journal of the American Statistical Association 96 (2001) 1077–1087.

[24] H. White, S. Boorman, R. Breiger, Social structure from multiple networks. i. blockmodels of roles and positions, American Journal of Sociology 81 (1976) 730–780.

[25] J. Daudin, F. Picard, S. Robin, A mixture model for random graph, Statistics and computing 18 (2008) 1–36.

[26] P. Latouche, E. Birmelé, C. Ambroise, Advances in Data Analysis Data Handling and Business Intelligence, Springer, 2009, Ch. Bayesian methods for graph clustering, pp. 229–239.

[27] M. Mariadassou, S. Robin, C. Vacher, Uncovering latent structure in valued graphs: a variational approach, Annals of Applied Statistics 4 (2) (2010) 715–742.

[28] P. Latouche, E. Birmelé, C. Ambroise, Overlapping stochastic block models with application to the french political blogosphere, Annals of Applied Statistics 5 (1) (2011) 309–336.

[29] E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, Mixed-membership stochastic blockmodels, Journal of Machine Learning Research 9 (2008) 1981–2014.

[30] E. Airoldi, D. Blei, E. Xing, S. Fienberg, Mixed membership stochastic block models for relational data with application to protein-protein interactions, in: Proceedings of the International Biometrics Society Annual Meeting, 2006.

[31] E. Airoldi, D. Blei, S. Fienberg, E. Xing, Mixed membership analysis of high-throughput interaction studies: relational data, 2007, ArXiv e-printsarXiv:0706.0294.

[32] B. Karrer, M. E. J. Newman, Stochastic blockmodels and community structure in networks, Phys. Rev. E 83 (2011) 016107.

[33] Y. Zhu, X. Yan, C. Moore, Oriented and degree-generated block models: Generating and inferring communities with inhomogeneous degree distributions, 2012, CoRR abs/1205.7009.

[34] X. Yan, J. E. Jensen, F. Krzakala, C. Moore, L. Shalizi, C.and Zdeborová, P. Zhang, Y. Zhu, Model selection for degree-corrected block models, 2012, CoRR abs/1207.3994.

[35] H. Zanghi, F. Picard, V. Miele, C. Ambroise, Strategies for online inference of network mixture, Annals of Applied Statistics 4 (2010) 687–714.

[36] P. Latouche, E. Birmelé, C. Ambroise, Variational bayesian inference and complexity control for stochastic block models, Statistical Modelling 12 (1) (2012) 93–115.

[37] C. Biernacki, G. Celeux, G. Govaert, Assessing a mixture model for clustering with the integrated completed likelihood, IEEE Trans. Pattern Anal. Machine Intel 7 (2000) 719–725.

[38] C. Biernacki, G. Celeux, G. Govaert, Exact and monte carlo calculations of integrated likelihoods for the latent class model, Journal of Statistical Planning and Inference 140 (2010) 2991–3002.

[39] J. S. Liu, The collapsed gibbs sampler in bayesian computations with applications to the gene regulation problem, Journal of the American Statistical Association 89 (1994) 958–966.

[40] H. Jeffreys, An invariant form for the prior probability in estimations problems, in: Proceedings of the Royal Society of London. Series A, Vol. 186, 1946, pp. 453–461.

[41] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 10 (2008) 10008–10020.

[42] J. Besag, On the statistical analysis of dirty pictures (with discussions), Journal of the Royal Statistical Society, Series B 48 (1986) 259–302.

[43] J. Hofman, C. Wiggins, A bayesian approach to network modularity, Physical Review Letters 100 (2008) 258701–259900.

[44] A. Mc Daid, T. Murphy, F. N., N. Hurley, Improved bayesian inference for the stochastic block model with application to large networks, Computational Statistics and Data Analysis 60 (2013) 12–31.

[45] N. X. Vinh, J. Epps, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, journal of Machine Learning Research 11 (2010) 2837–2854.

[46] E. Côme, E. Diemert, The noise cluster model, a greedy solution to the network community extraction problem, Information - Intelligence - Interaction 11 (1) (2010) 40–59.

[47] S. Fortunato, M. Barthélemy, Resolution limit in community detection., Proceedings of the National Academy of Sciences of the United States of America 104 (1) (2007) 36–41.